

Structural Directive

- ngIf
- ngSwitch
- ngFor

ngIf:-

- ngIf is a structural directive that can add or remove html element
 - ngIf works on the basis of true & false result of expression
 - If the condition is true the elements will be added on to the browser (DOM layout)
- Syntax: <h1 *ngIf="condition"> ... </h1>

app.component.html

```
<h1> Directives </h1>
<h1 *ngIf="false"> Angular </h1>
```

Output:-

Directives

Console Screen:-

<body>

<app-root ngHost ngVersion="8.2.10">

<h1>-ngContent-new-cc> directive</h1>

<!!--bindings=&

"ng-reflect-ng-if": "false" -->

Output:-

Directives

Angular

Console Screen:-

<h1>-ngContent-obj-cc>

directive </h1>

<!!--bindings=&

"ng-reflect-ng-if": "true" -->

app.component.ts

```

import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateURL: './template.html',
  styleUrls: ['./style.css']
})
export class AppComponent {
  title = 'directive';
  boolval = true;
}

```

Assigning value to ng-directive

app.component.html

```

<h1> directive </h1>
<h1 *ngIf="boolval">
  angular directive </h1>

```

Output:-

directive
Angular Directive

Inspect Screen

```

<h1 -ngcontent-exclude> directive </h1>
<!--bindings={}-->
  "ng-reflect-ng-if: "true" -->

```

If-else condition:-

Whenever we use else condition `<ng-template>` has to be used

Syntax:- `<ng-template #refername> </ng-template>`

app.component.ts

```

import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateURL: './template.html',
  styleUrls: ['./style.css']
})

```

title = "directive"

templateURL:

styleURLs:

}

export class AppComponent

{ title = "directive";

boolval = true;

}

app.component.html

<h1> Directive </h1>

<h1 *ngIf="boolval; else

elseBlock> If block will be executed </h1>

<ng-template #elseBlock>

<h1> else block will be

executed </h1>

</ng-template>

Output:-

directive

if block will execute.

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-component',
  templateUrl: './app-component.html',
  styleUrls: ['./app-component.css']
})
export class AppComponent {
  title = 'directive';
  boolval = false
}
```

app.component.html

```
<h1> Directive </h1>
<h1 *ngIf="boolval; else elseBlock> If block will be
executed </h1>
<ng-template #elseBlock>
<h1> else block will be ex-
ecuted </h1>
```

Output:- Directive

else block code will be executed.

ngSwitch:-

- ngSwitch is combination of two attributes. An attribute directive and structural directive.
- ngSwitch directive lets you hide or show html elements depending on an expression
- We can also define a default section by using ngSwitchDefault (if condition is not matching then default case could be executed)
- Three things ngSwitch, ngSwitchCase, ngDef, ngSwitchDefault.

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-component',
  templateUrl: './app-component.html',
  styleUrls: ['./app-component.css']
})
export class AppComponent {
  colon = 'blue';
}
```

app.component.html

```
<div [ngSwitch] = "colon">
<h1 *ngSwitchCase = "green">
  Green Content </h1>
<h1 *ngSwitchCase = "red">
  Red Content </h1>
<h1 *ngSwitchCase = "blue">
  Blue Content </h1>
</div>
```

Output:-

Blue Content.

app.component.ts Adding default case

```
import {Component} from '@angular/core'; app.component.html
@Component({
  selector: 'app-root',
  template: `
```

```
    <div [ngSwitch] = "color">
      <h1 *ngSwitchCase = "green">
        Green Content </h1>
      <h1 *ngSwitchCase = "red">
        Red Content </h1>
    <h1 *ngSwitchDefault>
      You have selected wrong color
    </h1>
  </div>
```

Output:- You have selected wrong color.

*ngFor:-

- ngFor is a structural directive.
- Its point is to repeat a given html template once for each value in an array.
- Syntax `*ngFor="let <value> of <collections>"`
- <value> is a variable, <collections> is a property on our component.

app.component.ts

```
import {Component} from '@angular/core';
@Component({
```

```
  selector:
```

```
  template: `
```

```
    <h1> ng for loop </h1>
```

```
    <div *ngFor = "let item of"
```

```
      names">
```

```
      {{item}}
```

```
    </div>
```

```
</h1>
```

```
</div>
```

```
</app-root>
```

```
</body>
```

```
</html>
```

app.component.html

```
  <h1> ng for loop </h1>
```

```
  <div *ngFor = "let item of"
```

```
    names">
```

```
    {{item}}
```

```
  </div>
```

Output:-

ng for loop

Vinat

Chahal

Sinaj